

**Tecnologie Web T**  
**17 Settembre 2019 – Compito**

**Tempo a disposizione: 3 ore**

La soluzione comprende la **consegna elettronica** dei seguenti file mediante l'apposito applicativo Web **esamix** (<http://esamix.labx>):

**Alberghi.zip** file zip contenente il sorgente java/class e pagine Web per punto 1  
**Booking.zip** file zip contenente pagine Web e file txt per punto 2  
**Pietanza.zip** file zip contenente il sorgente java/class e txt per punto 3

**Ogni file .zip consegnato DEVE CONTENERE TUTTI e SOLI i file creati/modificati e/o ritenuti importanti in generale ai fini della valutazione (ad esempio, descrittori, risorse statiche o dinamiche, codice Java e relativi .class, ecc.) e NON dell'intero progetto**

**N.B. Per superare la prova scritta di laboratorio ed essere ammessi all'orale, è necessario totalizzare almeno 18 punti (su un totale disponibile di 33), equamente distribuiti sui tre esercizi, ovvero almeno 6 punti sul primo esercizio, 6 punti sul secondo esercizio e 6 punti sul terzo esercizio**

---

**Studenti in debito di Tecnologie Web L-A**

**Viene richiesto lo svolgimento dei soli esercizi 1 (17 punti) e 2 (16 punti). Tempo a disposizione: 2 ore.**

**I 18 punti necessari per l'ammissione all'orale sono così distribuiti: almeno 10 punti sul primo esercizio e almeno 8 punti sul secondo**

---

**ESERCIZIO 1 (11 punti)**

Si realizzi un'applicazione Web, principalmente basata su tecnologie Java servlet, JSP e Javascript per la gestione delle **prenotazioni di n alberghi**.

L'applicazione Web deve gestire le disponibilità di camere per n alberghi; ogni albergo ha camere tutte uguali e, nel caso di prenotazioni per più giornate, il prezzo di una camera è uguale per tutte le giornate. I dati relativi agli alberghi vengono mantenuti server-side in un file di testo con n righe, una per ogni albergo gestito: ogni riga riporta *#ID\_albergo, #camere\_totali, prezzo\_STATICO\_camera\_per\_giornata*.

L'applicazione deve consentire a utenti non autenticati di selezionare *#ID\_albergo* e date di check-in e di check-out; si controlli, localmente al Web browser, che *#ID\_albergo* sia minore di n e che le date abbiano senso (check-out dopo check-in, check-in dopo data corrente). All'inserimento della data di check-out, senza ulteriore pressione di alcun tasto, si deve generare una richiesta che produce server-side il calcolo del prezzo dinamico della camera: il prezzo dinamico è uguale al prezzo statico maggiorato del 10% per ogni cliente che sta correntemente chiedendo informazioni su quell'albergo e per date almeno in parziale sovrapposizione. Una volta restituito il risultato, l'utente avrà la possibilità di finalizzare o meno la prenotazione; solo la finalizzazione decrementa il numero di camere disponibili. Per semplicità si consideri la possibilità di rappresentare le date come numeri naturali nell'intervallo [1, 365].

Infine, deve essere data la possibilità all'amministratore (previa autenticazione, pagina **admin.jsp**) di visualizzare lo stato complessivo delle prenotazioni finalizzate.

# Tecnologie Web T

## 17 Settembre 2019 – Compito

### ESERCIZIO 2 (11 punti)

Si realizzino le **pagine Web dinamiche** (basate su tecnologia **HTML**, **CSS**, e **JavaScript**) in grado di riprodurre il contenuto e il layout grafico dello “snapshot” del **sito Web Booking** di seguito riportato. Nello specifico, la pagina rappresenta il risultato della interrogazione riportata nella maschera arancione “Cerca”. La pagina presenta inoltre una sezione “Filtra per” che permette di raffinare la ricerca sulla base delle preferenze dell’utenza. Un menù orizzontale permette di visualizzare diverse categorie di risultati, ovvero “Scelte top per i viaggiatori business” (scelta di default evidenziata in verde), “Mostra prima le case”, “Prezzo (prima il più basso)”, “...”. La parte centrale della pagina è dedicata alla visualizzazione del risultato della ricerca, strutturato come lista di elementi composti. Ogni elemento composto è costituito da una immagine dell’albergo posta a sinistra (l’immagine deve essere un link al sito web dedicato dell’albergo) e da una sezione di testo che riporta ogni dettaglio relativo all’albergo come riportato in snapshot. **Lo studente è tenuto a giustificare adeguatamente le scelte relative ai principali costrutti HTML e CSS usati nella soluzione proposta nel file Booking.txt.**

**N.B.** La soluzione NON deve far uso del costrutto HTML frame. Per rappresentare le immagini riportate nello snapshot, si utilizzino figure di esempio a piacere.

Cerca

Destinazione/nome struttura:

Parigi

Check-in

5 giovedì 5 dicembre 2019

Check-out

7 sabato 7 dicembre 2019

Soggiorno di 2 notti

1 adulto

Nessun bamì

1 camera


☒ Viaggio per lavoro

Cerca

Parigi: 2.469 strutture trovate - tra cui 289 super occasioni!

Scelte top per i viaggiatori businessMostra prima le casePrezzo (prima il più basso)

Questi risultati includono anche sistemazioni condivise, come i posti letto in dormitorio. [Mostra solo camere private](#)



Appartamenti

**Blue Nights Apartments - Sébastopol**

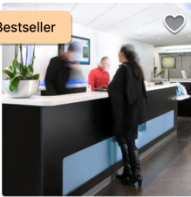
3° arrondissement, Parigi - Ottima posizione - [Mostra sulla mappa](#) - 1,4 km dal centro - Vicino alla metro

**Appartamento Standard** - 1 camera da letto • 1 zona giorno • 1 letto matrimoniale alla francese • 1 divano letto 27 m²

Buono 7,0  
275 recensioni

2 notti, 1 adulto  
€ 270  
include tasse e costi

Vedi disponibilità >



Bestseller

**ibis budget Paris Porte De Montmartre**

18° arrondissement, Parigi - [Mostra sulla mappa](#) - 5 km dal centro

Negli ultimi 10 minuti, 1 utente ha visto questa struttura per le tue stesse date

3 prenotazioni per le tue date da 1 ora a questa parte sul nostro sito

Super occasione

**Camera Tripla** - 2 letti singoli • 1 letto a castello

Punteggio medio 6,4  
12.758 recensioni

2 notti, 1 adulto  
€ 107  
include tasse e costi

Vedi le nostre ultime camere disponibili >

#### Filtra per:

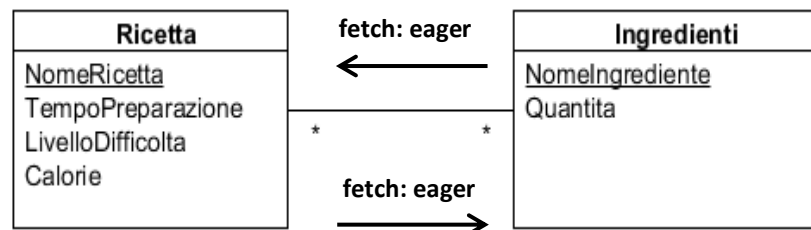
##### Preferiti dai viaggiatori per affari

<input type="checkbox"/> Bagno privato	2236
<input type="checkbox"/> Mostra solo le strutture disponibili	
<input type="checkbox"/> Colazione inclusa	1072
<input type="checkbox"/> Connessione WiFi gratuita	2317
<input type="checkbox"/> Bed & Breakfast	32
<input type="checkbox"/> Parcheggio	1063
<input type="checkbox"/> Hotel	1588
<input type="checkbox"/> 3 stelle	1047

2/3

**ESERCIZIO 3 (11 punti)**

Partendo dalla realtà illustrata nel **diagramma UML** di seguito riportato, si fornisca una soluzione alla gestione della **persistenza basata su Pattern DAO** in grado di **“mappare” efficientemente e con uso di ID surrogati** il modello di dominio rappresentato dai **Java-Bean Ricetta** e **Ingredienti** del diagramma UML con le corrispondenti tabelle relazionali derivate dalla progettazione logica del diagramma stesso.



Nel dettaglio, dopo aver creato da applicazione Java le tabelle all'interno del proprio schema nel database TW\_STUD di DB2 (esplicitando tutti i vincoli opportuni di PK e FK), implementato i **JavaBean** e realizzato le **classi relative al Pattern DAO** per l'accesso **CRUD** alle tabelle, si richiede la realizzazione dei seguenti **metodi**: (i) un metodo che restituisca gli ingredienti di *“Tagliatelle di nonna Pina”*; (ii) un metodo che restituisca i nomi delle ricette che contengono più di 5gr. di peperoncino.

Si crei poi un main di prova in grado di:

- (a) inserire due o più tuple nelle tabelle di interesse;
  - (b) fare uso corretto dei metodi realizzati ai punti precedenti al fine di produrre la stampa dei risultati sul file **Pietanza.txt**;
- il tutto, mediante opportuna **gestione delle transazioni**.

**N.B.** L'implementazione del Pattern DAO deve limitarsi al solo DBMS DB2. La soluzione Java deve sfruttare esplicitamente il mapping N-M specificato nell'UML. **Ogni ulteriore scelta da parte dello studente deve essere opportunamente giustificata con commenti nel codice.**